Development of a System for Content-Based Indexing and Retrieval in Large Image Databases

By

JOYSREE ROY

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JANUARY 1999

Development of a System for Content-Based Indexing and Retrieval in Large Image Databases

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of
MASTER OF TECHNOLOGY

by

JOYSREE ROY

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JANUARY 1999

22/9

CERTIFICATE

This is to certify that the work contained in this M Tech thesis entitled **DEVELOPMENT OF A SYSTEM FOR CONTENT BASED INDEXING AND RETRIEVAL IN LARGE IMAGE DATABASES** *has been carried out by* **JOYSREE ROY** *under my supervision and has not been submitted elsewhere for a degree*

Supervisor
Dr Sumana Gupta
Associate Professor
Dept of Electrical Engg
Indian Institute of Technology
Kanpur

Dedicated to

My Parents and Husband

Acknowledgment

I would take this opportunity to express my sincere gratitude towards Dr SUMANA GUPTA my thesis supervisor for her invaluable guidance and for giving me complete freedom in choosing a problem for the thesis work She has always been very patient and encouraging regarding the doubts and discussions during the thesis work I express my sincere respects to all my teachers who introduced me to different interesting subjects in the field of communications

I would like to thank my friends Arindam Naresh Ashok and Sanditshu who were helping me during the thesis work I would also like to thank Karthik Shiva Tasmin and Anil who were the support of my IIT career I am deeply indebted to my parents and my husband for their love and encouragement

My stay at IITK has been marked with many memorable moments During this period I was fortunate enough to listen to some of the eminent personalities from different fields From this environment apart from academics I got an exposure to different ideas regarding the various aspects of life, science and technology

Finally I thank Indian Government for giving me an opportunity to pursue the M Tech programme

Contents

List of Figures

Abstract

In this thesis work we are concerned with development of a content based indexing and retrieval system This is based on wavelet decomposition and quad tree segmentation Since the computational complexity has been one of the main barriers towards the use of similarity measures in large image databases we propose a hierarchical indexing scheme where computationally efficient features are used to subset the images before more sophisticated techniques are applied for precise retrieval Database systems which support indexing searching and retrieval have great demand Large visual database systems require effective and efficient ways of indexing and accessing visual data on the basis of content In our work we use 3 level wavelet transform to extract image features, significant features must first be extracted from image data in their pixel format Feature vectors of images are then constructed These feature vectors of database image segments are classified using clustering algorithm Hierarchical tree structure namely *Bucket PR k d* tree is used for efficient storage and searching of the clusters Content based image retrieval is performed by comparing the feature vectors of the query image and the mean feature vector of clusters With the large volume of visual data stored in a visual database image classification is a critical step to achieve efficient indexing and retrieval Our experiment illustrates that the proposed block oriented image representation offers a novel decomposition structure to be used to facilitate effective and efficient image retrieval

Chapter 1

Introduction

Digital images have become the medium of choice for describing storing and exchanging information in a variety of domains e g satellite images, biomedical images stock photo archives of news agencies etc With the decreasing cost and increasing performance of digital image/ video capture devices, computing power and storage capabilities, more and more visual information are now available on line in large image and video repositories However image and video data are much more voluminous than textual data The amount of data associated with visual information requires enormous storage capacity As a result image data compression techniques are used for reduction of the number of bits required to store or transmit images without any appreciable loss of information Among the image compression methods JPEG wavelet transformation and fractal coding [4] have recently drawn much attention Recent developments in the field of image data transformation and compression provide an interesting approach to describe the contents of an image [8]

Content based retrieval of imagery [3] [8] [9], [10] is presently an active area of research promising to provide powerful tools for database management in the near future Traditional methods of retrieving images from such a database rely on a number of descriptive keywords associated with each image These methods are inadequate in terms of effective description of the image contents because they are subjective(relative to the person indexing the images) Thus modern database systems should process images based on their content

Content based image retrieval has been proposed to allow retrievals to be performed on the basis of different aspects of the image content In this context a

challenging problem arises with many image databases in which queries are posed via visual or pictorial examples Such queries are termed as visual queries A common visual query to a image database system would involve finding all the images in that database which contain a subimage that is similar to a given query image Such retrievals must use embedded content features, such as the shape color, texture, layout and the positions of various objects in an image There are no generic tools which helps to understand the image content to a satisfiable extent The automatic retrieval of images on the basis of content thus pose difficult problems An approach which has drawn much attention recently involves the extraction of the color and texture features of images using image processing techniques and form image feature vectors based on the color and texture characteristics Content based image retrieval is then supported by searching and comparing the feature vectors of the query image with the database images Database systems which support indexing searching and retrieval of image are in great demand Different approaches adopted for these databases include

1 Multiresolution image representations for browsing and retrievals at various scales,

2 Textual description of data for keyword indexing,

3 Feature sets for searching by content e g , texture, color, shape spatial relationships, etc

Multiresolution(MR) image representations allow the user to browse through databases by viewing thumbnail sketches of image data But even with multiresolution browsing and retrieval, it is difficult to search through large amounts of data So content based approaches have been adopted which allow searching based on visual features of the image data In this process significant features are extracted from image data in their pixel format

There are many steps involved in the implementation of a feature-based retrieval system They are

1 Feature set identification,

2 Feature space compaction

3 Image segmentation and

4 Multidimensional point indexing

The important visual features of an image are described mathematically using feature sets that are derived from the digital data A content based search of the database proceeds by finding the items that are mathematically and visually similar to the query image Characterization of texture color and shape using feature sets

are being used in content based approaches for large image and video databases

Classification and discrimination [10] [4] of textures can be done based on the energies of image subbands Subband energy feature sets can be extracted from several typical image decompositions such as wavelet subband uniform subband, discrete cosine transform(DCT) and spatial partitioning In our present work we use Wavelet transforms for feature extraction Feature vectors of images are then constructed The subband energy based feature sets are applied to a system for indexing images by texture content in image databases, since the features can be extracted directly from spatial frequency decomposition of the image data These features are classified and indexed to assist efficient retrieval of image content For the efficient representation of database images we use an effective block oriented image decomposition structure which can represent image content in image database systems Content based image retrieval is performed by comparing the feature vector of the query image and the decomposed segments in database images

The rest of the report is organized as follows Chapter 2 describes image data segmentation and introduces the tree block oriented data structure namely quad tree In chapter 3 we discuss texture-based query and wavelet transform for image data compression and feature extraction This is followed by a description of a procedure to find the textural similarity between database images and the query image In chapter 4 we discuss the criteria used for grouping or clustering data followed by a procedure to store the clusters in a hierarchical tree structure namely bucket PR k d tree for easy and quick retrieval of database images for a given query Experimental results presented in chapter 5 Chapter 6 concludes the thesis and suggests the scope for future work

Chapter 2

Image Segmentation

Image segmentation refers to the decomposition of a scene into its components It is a key step in image analysis For example a document reader would first segment the various characters before proceeding to identify them The Figure below lists several image segmentation techniques

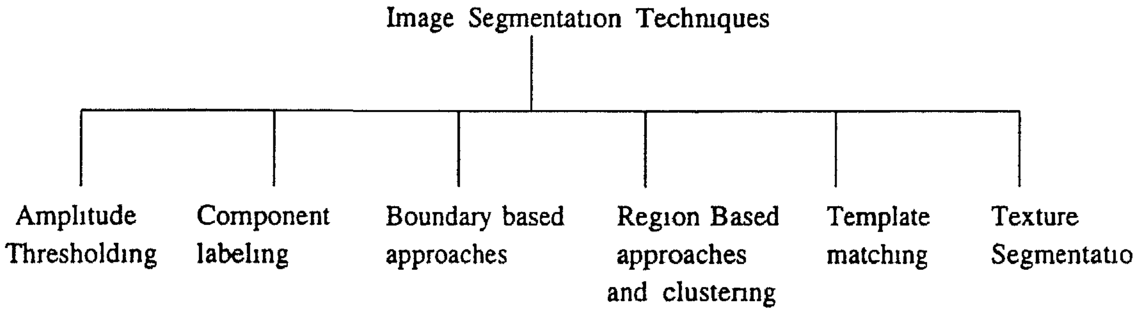


Figure 2 1 Different Segmentation Techniques

Amplitude thresholding is useful whenever the amplitude features sufficiently characterize the object The appropriate amplitude feature values are calibrated so that a given amplitude interval represents a unique object characteristic For example the large amplitudes of the remotely sensed visual and infrared (IR) images can represent low temperatures or high altitudes etc Component labeling is a simple and effective method of segmentation of binary images by examining the connectivity of pixels with their neighbors and labeling the connected sets Two practical algorithms used for connectivity analysis are Pixel labeling and Run length Boundary extraction techniques segment objects on the basis of their profiles Edge linking curve fitting etc are also applicable to image segmentation Template matching is a direct

method of segmenting an image in which an image is matched against templates from a given list The detected objects can be segmented out and the remaining image can be analyzed by other techniques This method can be used to segment busy images, such as journal pages containing text and graphics Texture segmentation becomes important when objects in a scene have a textured background In our work, we have used the region based approach to segmentation

2 1 Region-Based Approaches

As our main approach is to adapt the region based techniques we discuss this techniques here in detail The main idea of region-based segmentation techniques is to identify various regions in an image that have similar features Clustering techniques encountered in pattern recognition literature have similar objectives and can be applied to image segmentation

One class of region-based techniques involves *region growing* The image is divided into *automatic regions* of constant gray levels Similar adjacent regions are merged sequentially until the adjacent regions become sufficiently different

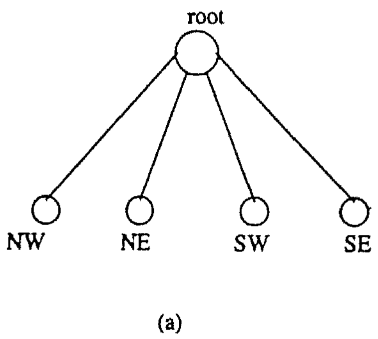
Instead of merging regions segmentation is done by splitting a given region For example , the image is split by *Quad-Tree* approach and then similar regions are merged Region based approaches are generally less sensitive to noise than the boundary-based methods In the next section we discuss the Quad tree decomposition

2 2 Quad-Tree Decomposition

As the feature vector of a database image may not correctly represent its subimsges the retrieval based on the comparison between the feature vectors of the query image and database images themselves may not provide satisfactory results for visual queries Thus image segmentation is a necessary step for effective searching of image databases Effective segmentation will isolate the important homogeneous regions and features of the images in the database, from which an index can be established for searching To avoid manual segmentation of images in a large database,a block oriented approach based on the quad-tree decomposition of images has been adopted

A quad tree [1] [2] [3] is a hierarchical image decomposition structure which can provide quick data access for image retrieval as hierarchical data structures have the

ability to focus on the interesting subsets of the data This focusing results in an efficient representation and improved execution time A quad tree is based on the principle of recursive decomposition of space The most studied quad tree approach to region representation termed as a region quad tree is based on the successive subdivision of the image array



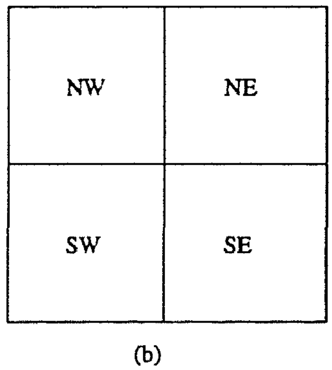


Figure 2 2 Quad Tree decomposition of an image array a) Tree representation of the image array b) Decomposed image array

Each decomposition of an image segment produces four equal sized quadrants Figure 2 2 demonstrates the positions of the four quadrants labeled in order NW, NE SW and SE within the decomposed segments The leaf nodes of the tree correspond to those blocks for which no further subdivision is necessary All non leaf nodes are said to be GRAY The tree decomposition and block decomposition are shown in figure 2 2

The image decomposition process using a quad tree structure can be described recursively, with the root representing the entire image and its children representing the decomposed segments, these, in turn become roots for further segmental decomposition Each internal node has exactly four children The original quad tree decomposition labels the decomposed segments white if they consist of white pixels only, black, if they consist of black pixels only and gray, if they consists of both black and white pixels Further decompositions is carried out only on gray segments Black and white segments remain unchanged

We have done the transformation and decomposition of an image array as follows, firstly, the entire image is decomposed using wavelet filtering then quad tree spatial blocks are obtained to represent appropriate regions in the original image array Then for these blocks we take wavelet transformation to obtain the feature vector of each

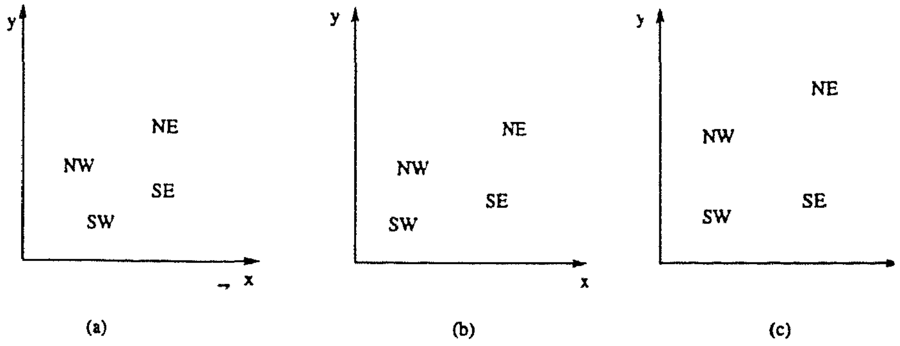
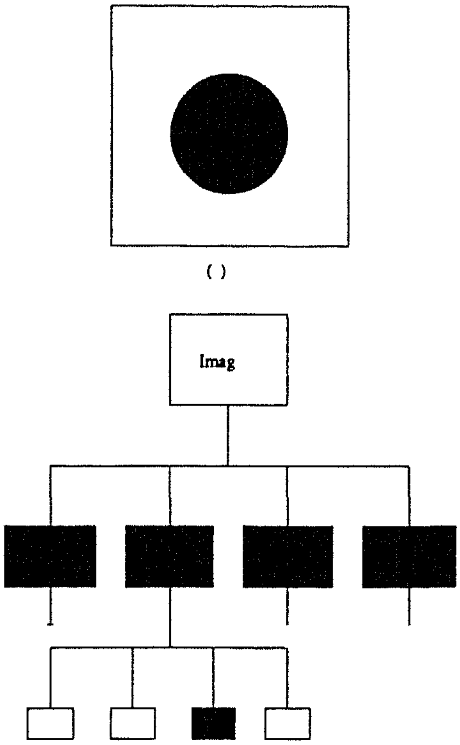


Figure 2 3 Splitting examples (a) NO Split all children belong to single class (b) Split into two children NW and SW form a texture class and children NE and SE form a second texture class (c) Split into four four texture classes present in this spatial block

block In this way, at each level we get the feature vector for each segment of the image array But the procedure we followed to proceed for next level segmentation is slightly different We examine at each level the merging condition of the blocks using a distance threshold that is discussed in the next chapter

Now we discuss the procedure that we follow to combine the image data compression or transformation and the image data segmentation Here we have implemented a changed definition of leaf nodes Before the four children are spawned by each parent condition of merging are tested The distance threshold is computed for each child on the basis of the extracted texture features The Euclidean distances in the feature space are measured from the parent node to each child If the distances to all four children fall within the threshold of the corresponding child, a single texture would be declared in the parent node and no further decomposition is necessary Otherwise pairwise grouping of the children are performed That is, if the Euclidean distance between two neighboring children falls below their respective thresholds, the children are merged as a single child Different situations may occur depending on the distance between nodes This is shown in figure 2 3 The quad tree decomposition is then iterated on each child So the quad tree is grown iteratively by testing the condition for splitting a parent block based on the texture content of children The quad tree decomposition can be characterized as a variable resolution data structure

An example of a quad tree decomposition is given below in figure 2 4



(b)

Figure 2 4 a) Original image b) Quad tree representation of the image

Chapter 3

Feature Extraction in Texture-based Image Query

Given the large volume of image data collected in a visual database, a manual approach for detecting and classifying images is both highly inefficient and prone to errors arising from subjective interpretation An approach which has drawn much attention recently involves the extraction of the texture features of images from their mathematical representations used in various image compression techniques

The visual characteristics of homogeneous regions of real world images are often identified as texture These regions may contain unique visual patterns or spatial arrangements of pixels which regional gray-level or color alone may not sufficiently describe Typically, textures have been found to have statistical properties structural properties or both Texture is an important underlying primitive in human visual perception including coarseness contrast and directionality These texture characteristics can be used to support content based image retrieval

Texture is used to describe content of many real world images for example, clouds, trees bricks, hair,fabric all have textural characteristics Because of the fundamental importance of texture information for human vision texture can provide a meaningful tool for searching image databases By indexing on the texture contents of images in the database a user may search through large volumes of images using texture keys Ideally there is no restriction that the texture belongs to predefined classes Furthermore the Query by texture" can be combined with other descriptions of color and shape to formulate overall image content based queries

Towards the development of Query by texture", it is necessary to find a mean

ıngful measure of the sımılarıtʏ of textures(texture dıscrımınatıon) and to develop procedure for segmentıng ımages based on textual content(texture segmentatıon) Furthermore dıscrımınant functıons are needed to gauge the sımılarıty between the texture regıons and the texture key used for searchıng Lıkewıse the texture dıscrım ınant functıons can be used to match blocks wıthın each ımage to produce the texture segmentatıon The goal of thıs segmentatıon ıs to provıde ındexıng of ımages ın the database Segmentatıon ıs successful when ımportant regıons of texture of the ımages are obtaıned Usıng the spatıal quad tree approach each quad tree node poınts to a block of ımage data Chıldren nodes are merged when the dıscrımınant functıons ındıcate that the chıldren blocks contaın suffıcıently sımılar textures A ' Query-by texture' examınes the blocks ıdentıfied by the final quad tree structures to test the sımılarıty to a texture key used for the search

3 1 Textural Sımılarıty

The Brodatz texture collectıon(Brodatz ımages) was used to obtaın dıscrımınant functıons(equatıon 3 1) Thıs procedure, used ın our work, constructs lınear com posıtes of the features whıch provıde maxımum average separatıon among traınıng classes The Mahalanobıs dıstance [15] whıch ıs the squared dıstance ın the trans formed feature space was used to measure the sımılarıty between textures Thıs ıs gıven by equatıon 3 2 In ordınary classıficatıon of textures or comparısons of many textures the relatıve rankıngs of the Mahalanobıs dıstances are used to ıdentıfy clos- est matches Subsequently ın response to a ' Query by texture all textures ın the database wıll be sorted by thıs dıstance to the texture-key Thıs search needs no threshold to be establıshed to determıne when textures are no longer "sımılar ' How ever, to decıde whether two textures are sımılar or not as ın quad tree segmentatıon requıres a threshold ın dıstance

Usıng a fixed dıstance threshold for determınıng whether two textures are e suf fıcıently sımılar wıll not be optımal for all types of textures Dependıng on the characterıstıcs captured by the extracted feature sets the wıthın class varıance for vısıbly sımılar textures wıll vary from class to class The dıstance threshold depends heavıly on the block sıze from whıch the features are extracted and on the energy of the feature set

There ıs a relatıonshıp between a block sıze and the estımated feature probabılıty dıstrıbutıon of the block Sınce smaller texture blocks wıll contaın fewer data poınts

from which to derive statistical features there will be a larger deviation in features extracted from these blocks of similar textures This results in a greater variance in distance between possibly similar textures necessitating a higher distance threshold for comparing textures of smaller block size Likewise features extracted from larger blocks produce smaller within class variation necessitating a lower distance threshold for comparing textures of larger block size

The distance threshold can be determined from information about the textures that are compared namely image region size and energy of the feature set Performing a linear regression analysis on this data, the threshold function is given as

$$th = 82\ 057 * ||fv||/s - 1\ 748 \tag{3 1}$$

Here th is the threshold to be used in the Mahalanobis distance, $||fv||$ is the energy of the transformed feature vector and s is the number of pixels in the image block

3 2 Searching Quad-Trees

After decomposing the image array in a quad-tree we search through the tree to find out the matching between the query image and database image Each node of the tree contains the feature vector of a segment rather than the original pixel data Also the feature vector is generated only on the basis of the pixel data rather than the entire image

A relationship between a given query image and any image in the database is determined by comparing the feature vector of query image with different portions i e different nodes of the database image until a match is found This is done by traveling through the quad tree of the database image from the root to the bottom We use the root mean square metric to compare the distance between the feature vector of the query image and that of the image segments of the database images For the given feature vectors $v1(a_1, a_2, \quad , a_n)$ and $v_2(b_1 \quad , b_n)$, of database image and query image respectively, their distance is computed as follows

$$dist(v1, v2) = \sqrt{(1/n \sum_i (a_i - b_i)^2)} \tag{3 2}$$

For a given threshold if the distance between the query image and a segment of a database image is less than the threshold, then we select this database image as a matched image

But this direct search is not efficient as using this we have to search a large number of nodes So in next chapter we discuss the clustering or grouping of feature vectors of the nodes of the quad tree representation of the database image

3 3 Wavelet Transforms

In computer vision it is difficult to analyze the information content of an image directly from the gray level intensity of the image pixels Indeed, this value depends upon the lighting conditions Generally the structures to be recognized have very different sizes Hence, it is not possible to define a priori an optimal resolution for analyzing images To process the image at different resolution, image information can be reorganized into a set of details appearing at different resolutions A multiresolution decomposition enables us to have a scale invariant interpretation of the image A multiresolution representation [11] provides a simple hierarchical framework for interpreting the image information At different resolutions the details of an image generally characterize different physical structures of the scene At a coarse resolution, these details correspond to the larger structure which provide the image "context" It is therefore natural to analyze first the image details at a coarse resolution and then gradually increase the resolution Such coarse-to-fine strategy is useful for low level image processing such as stereo matching and template matching

Multiresolution representations are very effective for analyzing the information content of images the different of information between the approximation of the signal at the successive resolutions can be extracted by decomposing this signal on a wavelet orthogonal basis This decomposition defines an orthogonal multiresolution representation called a wavelet representation It is computed with a pyramidal algorithm based on the convolutions with quadrature mirror filters and this is why this computation is efficient

Like the first Fourier transform(FFT) the discrete wavelet(DWT) is a fast linear operation that operates on a data vector whose length is a power of two transforming it into a numerically different vector of the same length Also like the FFT, the wavelet transform is invertible and in fact orthogonal the inverse transform, when viewed as a big matrix, is simply the transpose of the transform So both FFT and DWT, therefore can be viewed as a rotation in a function space, from the input space(or time) domain to a different domain

There are many possible sets of wavelets A particular set of wavelets is specified

by a particular set of numbers called wavelet filter coefficients Here a class of wavelet filters discovered by Daubechies are used by us This class includes members ranging from highly located to highly smooth The simplest and most localized member often called DAUB4 has only four coefficients, c0 c1 c2 c3

The DWT consists of applying a wavelet coefficient matrix hierarchically, first to the full data vector of length N then to the smooth vector of length $N/2$ then to the smooth smooth" vector of length $N/4$ and so on until only a trivial number of Smooth Smooth components remain This procedure is called pyramidal algorithm The output of the DWT consists of these remaining components and all the "detail components that were accumulated along the way

A wavelet transform of a 2-dimensional array is easily obtained by transforming the array sequentially on its first index(for all values of its other indices) then on its second and so on Each transformation corresponds to multiplication by an orthogonal matrix

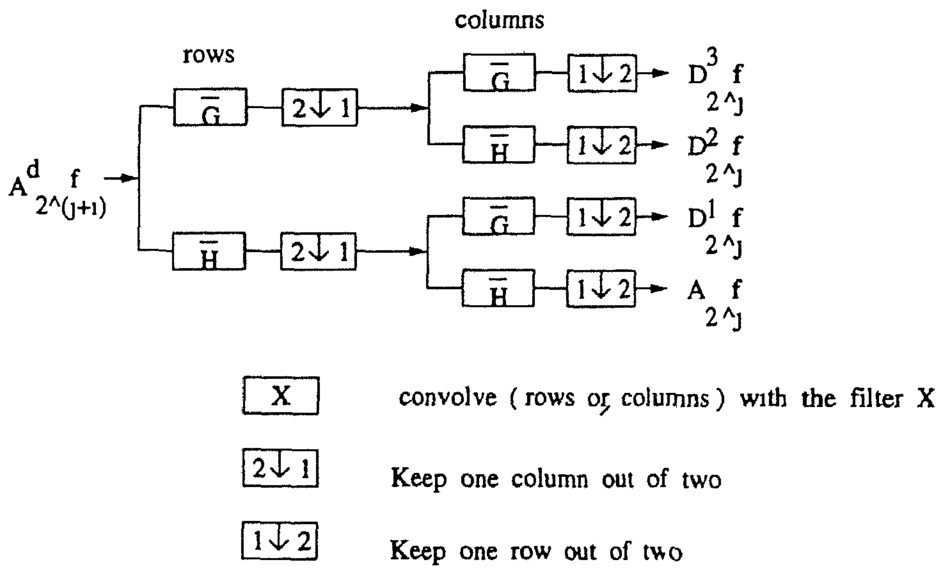


Figure 3 1 Decomposition of an image $A^d_{2^{j+1}} f$ into $A^d_{2^j} f$ $D^1_{2^j} f$, $D^2_{2^j} f$ and $D^3_{2^j} f$ This algorithm is based on the one dimensional convolution of rows and columns of $A^d_{2^{j+1}} f$ with the one-dimensional quadrature-mirror filter G and H

An immediate application of the multidimensional wavelet transform is in image data compression For images the wavelet representation differentiates several spatial orientations The two-dimensional wavelet transform can be seen as one dimensional wavelet transform along the x and y axes Wavelet based subband coding allows

simultaneously for high spatial resolution at high spatial frequencies and high spatial frequency resolution at low spatial frequencies Thus a filter bank based on wavelets could be used to decompose the image into low pass and high pass spatial frequency bands In two dimensional wavelet transform at each step we decompose $A^d_{2^{j+1}}f$ into $A^d_{2^j}f$ $D^1_{2^j}f$, $D^2_{2^j}f$ and $D^3_{2^j}f$ This algorithm is illustrated by a block diagram in Figure 3 1 We first convolve the rows of A f with one dimensional filter retain every other row convolve the columns of the resulting signals with another one dimensional filter and retain every other column The filter used in this decomposition are the quadrature mirror filter(QMF) Wavelet transforms are closely related to tree structured filter banks and hence to multiresolution analysis Tree structured filter banks give rise to nonuniform filter bandwidths and nonuniform decimation ratios in the subbands These two non uniformities are the fundamental ingredients of the wavelet transform Wavelet based subband coding allows simultaneously for high spatial resolution at high spatial frequencies and high spatial frequency resolution at low spatial frequencies Thus a filter-bank based on wavelets could be used to decompose the image into low pass and high pass spatial frequency bands

Orthogonal wavelet transforms provide interesting insights on the statistical properties of images Each of the subbands obtained after filtering have uniform texture information Feature extraction can be performed by energy estimation in subbands The feature used are computed from the mean absolute value and variance measures on the subbands produced from three iterations of QMF wavelet decomposition The advantage of these wavelet spatial frequency approaches is that simple statistics computed from the subband images are used as the subbands have limited spatial information Wavelet representations also provide a multiresolution structure and energy compaction to enable compression

We have used wavelet transformation in our work The wavelet decomposition and feature extraction processes are no longer independent for each spatial block At first, the entire image is decomposed using wavelet filtering then quad tree spatial blocks point to appropriate regions in the full original image Wavelet transform is applied to these regions to extract the feature information and construct the feature vectors for respective blocks This is shown in figure 3 2

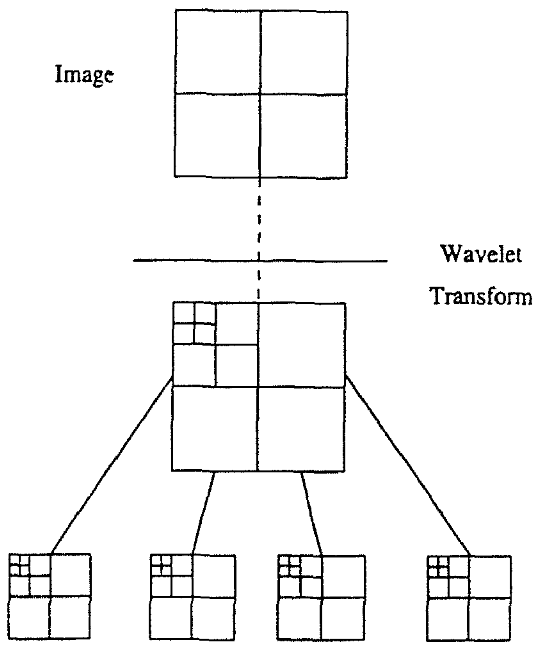


Figure 3 2 Quad tree blocks points to the appropriate regions in the full original image for feature extraction

example data may crowd into several partitions while other remain sparse or empty This result is nonuniform search performance

4 1 2 Data Clustering

An improvement over partitioning the feature space is to gather data points into clusters In unsupervised learning we attempt to identify clusters or natural groupings in the feature space A cluster is a set of points in the feature space with local density that is large(relative maximum) compared to the density of the feature points in the surrounding region As we have feature space which contains vectors whose components are real numbers, a feature class can be characterized by its clustering properties in the feature space Clustering techniques are useful for image segmentation and for classification of row data to establish classes and prototypes Clustering is also a useful vector quantization technique for compression of images

The feature space search can be broken into two stages The first will identify the appropriate cluster and the second will search all the data points in that cluster Search performances at each stage can be regulated by the clustering algorithm For example, a bound set on the maximum number of data points per cluster will produce a bound on the second stage of the search algorithm This results in more consistent performance than partitioning the feature space

The success of clustering techniques rests on the partitioning of the feature space into cluster subsets A general clustering algorithm is based on split and merge ideas shown in figure 4 1 below

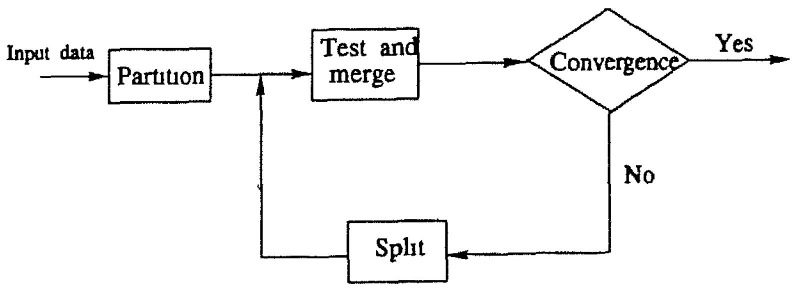


Figure 4 1 A clustering approach

If the vectors are characterized by clusters which are far apart simple recognition schemes such as minimum distance classifiers where the features are classified by distance functions, may be successfully employed The motivation for using distance functions as classification tool follows naturally from the fact that the most obvious

way of establishing a measure of similarity between feature vectors which we consider as points in Euclidean space is by determining their proximity

The method of feature classification by distance functions can be expected to yield practical and satisfactory results only when the feature groups or clusters do not overlap When the clusters overlap it becomes necessary to utilize more sophisticated techniques for grouping the feature space Overlapping clusters are the result of a deficiency in observed information and the presence of measurement noise Hence the degree of overlapping can often be minimized by increasing the number and quality of measurements performed on the feature vectors

As the database consists of a number of images so searching of individual quad tree which represents a database image may be highly time consuming This problem is circumvented by a novel clustering and indexing technique on the feature vectors generated from the image segments Before indexing the database images features vectors are grouped depending on different criterion

Data grouping is not a rigorously or uniformly defined concept Some speak of it as the subdivision of a set of data into subsets whose members are similar in some sense Others seek subsets exhibiting a cohesiveness which is not entirely related to point to-point similarity In the most general case, the proper number of subsets is unknown Here we consider a case where the number of subsets is fixed Many procedure for solving the general problem combine an algorithm for grouping data into a fixed number of subsets with some procedures for merging or splitting subsets

Two basic techniques for grouping data appear in the literature

1 Hierarchical approach

2 Direct approach

In hierarchical approach, data is divided into subsets and each subset is divided into further subsets and so on until the desired number of subsets is created The direct approach, on the other hand is to subdivide the data into the desired number of subsets in one step The direct approach seems to yield more reasonable classification but require more computation Hierarchical classification may require only the ability to divide data into two subsets

Data grouping is often accomplished with the aid of a criterion This criterion assigns a number to each possible partition of the data The partition which yields an extreme value of the criterion is chosen as the desired partition The criterion must meet two basic requirements

Performance The resulting partition must fall along natural boundaries of the data when such boundaries are well defined

Efficiency There must exist an efficient algorithm for finding optimum partition

Another requirement is uniqueness We would like to have one partition which extremizes our criterion Although uniqueness is an important factor, we consider here only performance and efficiency

We discuss in this chapter the criterion to be considered for grouping a basic algorithm for clustering the data and then define a linear transformation [13] which can be applied to the original data to improve the performance of the clustering algorithm

4 2 Criteria and Algorithms for Grouping Data

The criterion which we consider here can be defined in terms of the statistical scatter matrices introduced by Wilks [19] Let X_1 X_2, $, X_n$ a set of L-dimensional column vectors, be the data set We wish to divide this set into M groups G_1 G_2 $, G_M$ with populations N_1 N_2 N_M respectively The scatter matrices are defined as follows

Total scatter

$$T_L \triangleq \sum_{k=1}^{N} X_k X_k^T \tag{4 1}$$

Intragroup scatter

$$W_j \triangleq \sum_{X_k \in G_j} (X_k - C_j)(X_k - C_j)^T \tag{4 2}$$

where

$$C_j = \frac{1}{N_j} \sum_{X_k \in G_j} X_k \tag{4 3}$$

Total intragroup scatter

$$W \triangleq \sum_{j=1}^{M} W_j \tag{4 4}$$

Intergroup scatter

$$B \triangleq \sum_{j=1}^{M} N_j C_j C_j^T \tag{4 5}$$

The superscript T denotes transposition It is easily shown that

$$T_L = W + B \tag{4 6}$$

regardless of the partition Another important property of the scatter matrices is the fact that the eigenvalues of $W^{-1}B$ denoted by λ_1 λ_L are invariant under nonsingular linear transformations of X_k s

Criteria for Grouping

Several criteria may be defined in terms of scatter matrices defined above The intragroup scatter is a measure of the degree of association between the members of the group The criteria we consider here is based on some measure of the magnitude of W, the intragroup scatter

The criterion J_0 is defined as

$$J_0 = trW = \sum_{j=1}^{M} \sum_{X_k \in G_j} \|X_k - C_j\|^2 \qquad (4\ 7)$$

where trW is the trace of the matrix W J_0 may be interpreted as the mean squared distance to the group center The optimum partition is taken as the one which minimizes J_0

J_0 is not invariant under nonsingular linear transformations of the X_k s This means that by changing the coordinate system of the original data, one may alter the optimum position A criterion of the form

$$J = f(\lambda_1 \quad , \lambda_L) \qquad (4\ 8)$$

is invariant under nonsingular linear transformations Here the optimum partition is the one which maximizes the criterion

A basic assumption in clustering is that the group structure is somehow reflected by the data If we feel that this structure is reflected in an invariant manner, then we should use an invariant criterion to detect it This was essentially what Friedman and Rubin [20] assumed and the notion was supported by their experiments

4 3 Algorithm

The basic algorithm shown in Fig 4 2 has been widely used and in our case, we have used this algorithm This algorithm yields a grouping of the data which minimizes, at least locally J_0

J_0 is good in terms of the efficiency requirement If the coordinate system is suitably chosen J_0 becomes a function of $\lambda_1, \quad , \lambda_L$, thereby becoming invariant to any nonsingular linear transformation on the vector space From our definitions it follows that if $N \geq L$ the total scatter matrix T_L is a positive definite matrix Then, there exists a nonsingular matrix A, such that

$$AT_LA^T = I \qquad (4\ 9)$$

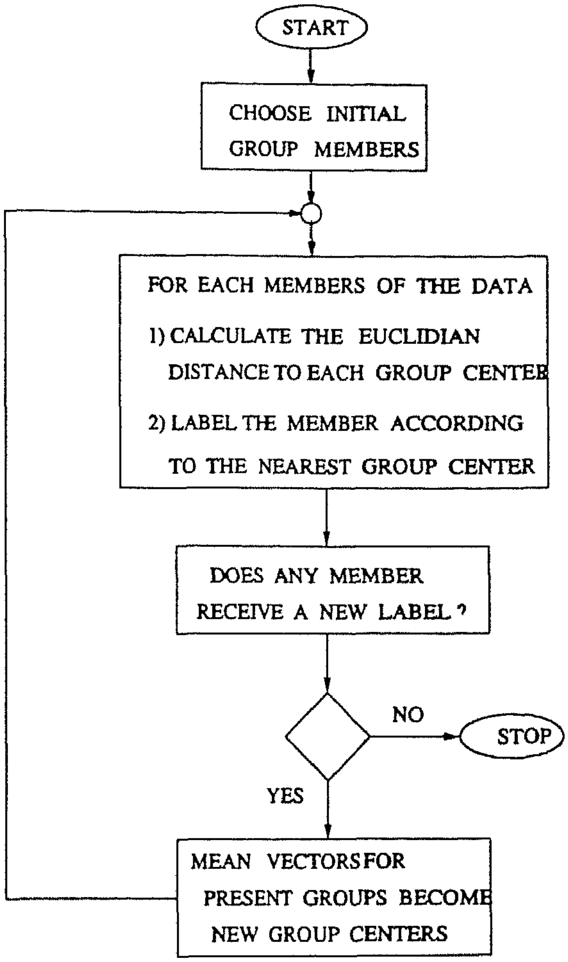


Figure 4 2 Basic grouping algorithm

regardless of the partition In fact as T_L is positive definite the matrix A is actually the eigen vector matrix of the total scatter matrix T_L The nonsingular linear transformation A diagonalises the total scatter matrix and makes J_0 a function of $\lambda_1, \quad, \lambda_L$, which takes the form

$$J_0 = \sum_{i=1}^{L} \frac{1}{1+\lambda_i} \tag{4 10}$$

thereby making the criteria invariant to any nonsingular linear transformation The application of this nonsingular linear transformation to the data set before actually starting the basic grouping algorithm improves the performance of the clustering algorithm

4 4 Formation of Hierarchical Data Structure for Cluster Storage and Retrieval

Using the clustering algorithm mentioned above, we get two clusters at each stage As there is an upper bound on the maximum number of data points per cluster, the pair of clusters formed at each stage may require further sub-clustering depending on the number of data points in that particular cluster All these clusters and sub-clusters need to be represented in the form of an appropriate hierarchical data structure for proper storage and later on for quick and easy retrievals of the matches to a given query image For this purpose, we have used a spatial data structure similar to a *Bucket PR k d tree* [21] as shown in Figure 4 3 A k d tree typically handles a point query determining if a given data point is in the database and if so yields the information corresponding to the record in which it is stored In the term *k d tree* *k* denotes the dimensionality of the space being represented Typically, it is a binary search tree with the distinction that at each depth a different attribute or key value is tested when determining the direction in which a branch is to made

k-d trees are useful in applications involving search Let us consider a typical query that seeks all nodes within a specified distance of a given point The k d tree structure serves as a pruning device on the amount of search required, i e many nodes need not be examined Assuming a k-dimensional space at each node, only one key needs to be compared instead of k In our case this key is the Euclidean distance between the query data point and the mean of the feature vectors of the two clusters Terminal nodes, as well as other nodes do not occupy too much space since the number of pointers per node is reduced to 2 for all values of k instead of 2^k

Apart from the point based trees, e g k d tree there are a number of ways of adapting the region based trees, like MX quad trees [21] PR quad trees [21], to represent point data The MX quad tree is feasible as long as the domain of data points is discrete and finite If this is not the case, the data points cannot be represented using this because the minimum separation between data points is unknown This observation leads to an alternative adaptation of the region quad tree to point data which associates data points(which need not be discrete) with quadrants typically We call this *PR quad tree*(where P stands for point and R stands for region)

In our work, we describe k d tree as an analogous data structure that uses binary trees instead of quad trees Such a data structure can also be termed as a PR k d tree or a PR bin tree

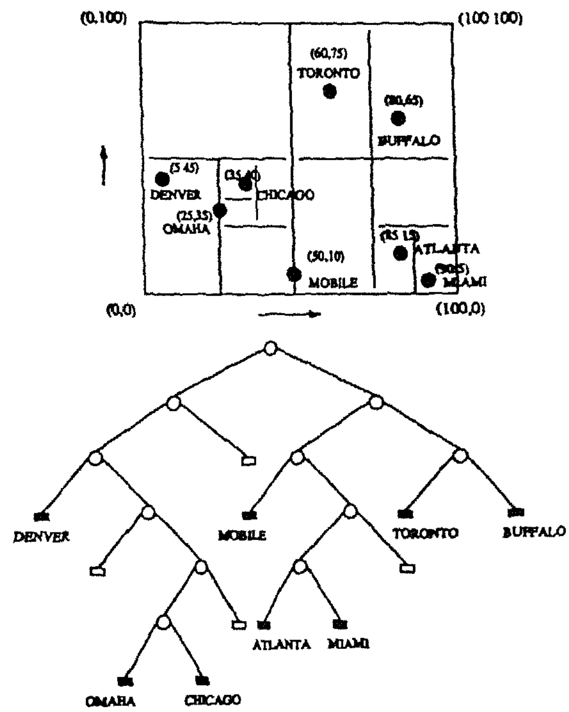


Figure 4 3 PR k d tree and the records it represents

Using standard tree structures for image databases is a bit impractical since a tree structure requires that pointers be followed and a point query results in the retrieval of the closest matching point in the database If we have similar image segments present in different parts of the same image or in multiple images present in the same database, and if we give a similar query image, we would like to retrieve all matching images from the database This is not possible in a PR k-d tree since each quadrant or cluster can contain at most one data point To overcome this problem, methods have been designed to collect the points into sets(termed Buckets) Then the access of these buckets is organized by the use of an appropriate tree to speed up the process of image retrieval We term such techniques as *"bucket methods"* Having buckets of capacity $c(c > 1)$ reduces the maximum dependence of the PR k-d tree on the minimum Euclidean distance separation of two distinct points to that of two sets of c points The term Bucket PR k d tree is used to refer to this structure [21] In our case unlike bucket PR k-d tree which divides the data set into quadrants in k-dimensional vector space, we form two sub-clusters at any stage, in the k dimensional vector space, from the existing points in the cluster under consideration This is carried out as long as it *overflows* i e the number of data points in the cluster is greater than the bucket size The process, of sub-cluster formation continues until all the terminal nodes

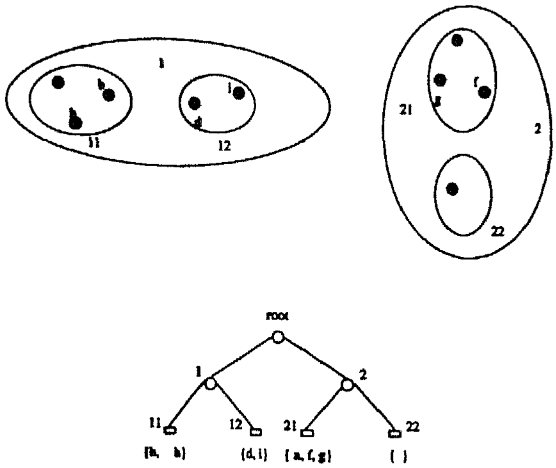


Figure 4 4 Two clusters and sub-clusters with their tree representation

have partially filled bucket or at most full buckets In Figure 4 4 we have shown an example of the clustering process using the hierarchical tree structure i e bucket PR k d tree First the feature space is divided into two clusters, then within each cluster sub-clustering is done Each group contains data points that is less than or equal to the bucket size The tree representation of this clustering process

For a given query image, retrieval of relevant information is done in a standard way as in the PR k d tree In the present work, the only difference is that , the query causes the retrieval of all the data points in the bucket corresponding to the terminal node to which the query data points has been guided to and present them in a suitable manner depending on their proximity to the query data points

Chapter 4

Indexing The Database Images

4 1 Hierarchical Searching

As the number of images and texture regions increase, it becomes necessary to utilize a searching procedure through the database more efficient than exhaustive search The ordering of the discriminant functions by significance is a natural by product of the feature space compaction performed by the Fisher Discriminant Analysis [10] The energy compaction may be exploited by adapting a hierarchical approach to searching Therefore, each dimension of the feature space is searched independently in order of significance For example a binary search may be conducted on sorted feature data beginning with the most significant discriminant functions At first, the candidate match list may be kept larger than the final desired number of retrievals As the list is subsequently searched using the remaining discriminant functions it may be further truncated This technique of successive refinement reduces the complexity of searching the high dimensional feature space by reducing the overall number of comparisons

4 1 1 Feature Space Partitioning

A second alternative to searching the high dimensional texture feature space is to partition the feature space This way the search can be conducted in two stages The first will identify the appropriate partition in feature space The next stage will examine only the data points in the partition However, if the dimension of the feature space is large, it is unlikely that uniform partitioning will be optimal For

Chapter 5

Experiments

A testbed of the database has been constructed from 15 Brodatz texture images The test images are 128*128 pixels in size The query images were chosen as random subimages of randomly chosen Brodatz images of size 32*32 pixels

Database images were decomposed using the quad tree decomposition procedure as explained in chapter 2 Feature extraction was performed on all image segments in these trees using wavelet transform which filter out texture features of images

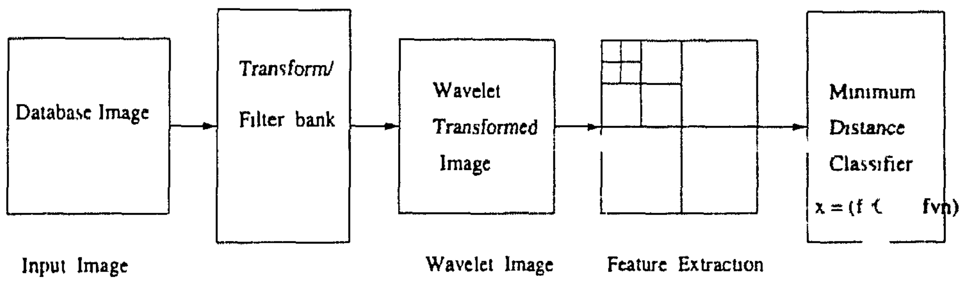


Figure 5 1 Texture classifier for Brodatz textures using subband energy based features

Three levels of subband decomposition were used in our experiments Each of the subbands obtained after filtering have uniform texture information In our experiment feature extraction was performed using Daubechies 4 wavelet filter Each three level transformation produces 10 subbands or image blocks as shown in Figure 3 1 in chapter 3 Features were extracted by energy estimation in subbands Two energy features mean and variance were computed for each subband i e for each decomposition energies were measured by calculating the variance and mean abso-

lute value of each subband Thus each feature vector has 20 elements This feature extraction procedure was applied to all image segments in the tree representation of each image Each node in the tree thus represents a segment of the image and its feature vector The feature space contains all generated feature vectors of image segments A block diagram representation of the feature extraction process is presented in Figure 5 1

Database image segments were then grouped using the clustering algorithm discussed in chapter 4 (Figure 4 4) To get proper clusters we used a linear transformation on feature vectors in the feature space [21] Then we constructed a hierarchical tree structure named Bucket PR k d tree (Figure 4 3) for cluster storage and easy and quick retrieval The feature vector for each query image was calculated using the root mean square(*rms*) metric This was used to compare the distance between the feature vectors of the query image and the mean feature vectors of each node of the bucket PR k d tree At the end of the search procedure when a bucket is reached the database image segments which fall in that particular bucket are retrieved as matched image segments In this way we get the information about the database images which contain a segment/segments which is/are similar to the query image The flow chart given in Figure 5 1 explains the steps we followed in our experiment

Database images belonging to different texture groups but containing closely similar texture features may fall within a very small distance Also some semantically irrelevant images may have feature vectors that fall within a very small distance To prevent the retrieval of such irrelevant images clustering approach is needed to classify the images into different categories before retrieval is performed

5 1 Result

Results of the query are shown for two query images

Figure 1 of the result shows four matched database images Each of them contains a segment that is similar to the query image Table 1 shows the matched image segments for this query The second column shows the name of the matched database images and the third column shows the *rms* distances between the query image and the corresponding database images Database images or image segments belonging to different texture groups but containing closely similar texture features may fall within a very small distance As indicated in this Table some irrelevant segments were retrieved as matched segments of the database images This means that though

they belong to two different texture groups in Brodatz texture images that were used to construct our database the texture features in the different segments contain closely similar texture features The retrieval result also depends on the bucket size If the bucket size is too small some relevant information might not be retrieved On the other hand, if the size is too big there is a possibility of getting some irrelevant information So we kept the bucket size to an optimum value 10 Referring to the distance column in Table 1 we observe that distances for the irrelevant segments are higher than that of relevant segments Image segments are sorted here according to their distances with the query image

Figure 2 of the result shows four relevant images Table 2 shows only the relevant segments of these images

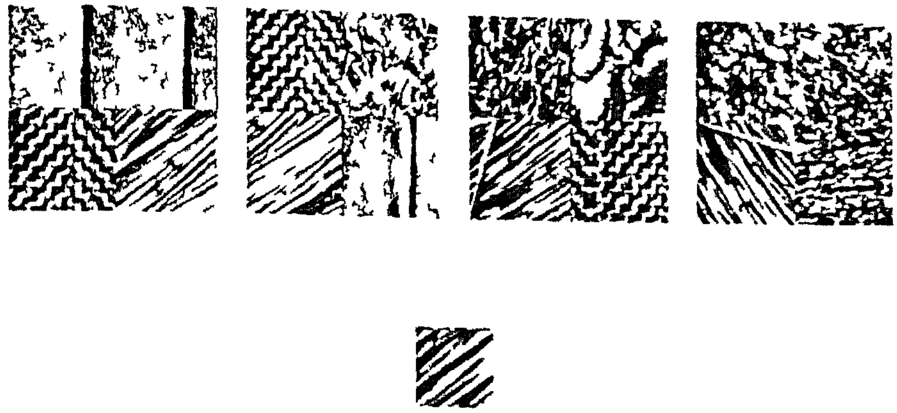


Figure 1 Name of the matched database images in order db9 pgm db1 pgm db10 pgm db7 pgm

Code	*File*	*Distance*
04	db9 pgm	56676 000000
03	db1 pgm	60525 000000
03	db10 pgm	68682 000000
03	db7 pgm	72366 000000
04	db10 pgm	77557 000000
03	db9 pgm	96805 000000
04	db7 pgm	100689 000000
01	db1 pgm	104481 000000
0113	db1 pgm	132084 000000
0112	db1 pgm	161534 000000

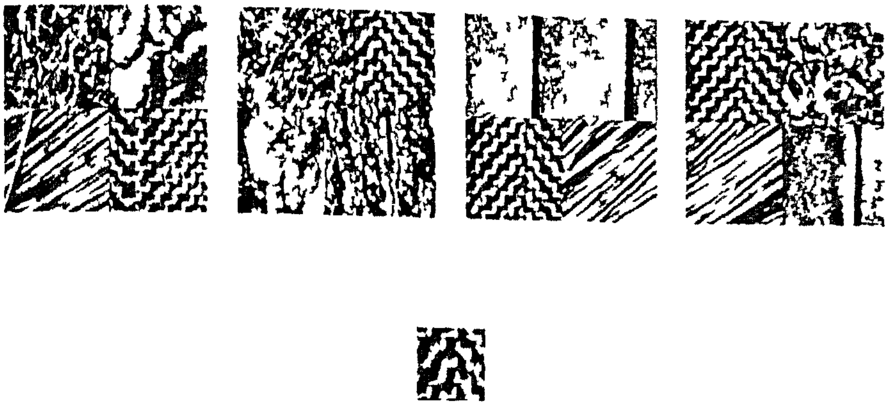


Figure 2 Name of the matched database images in order db10 pgm db8 pgm, db9 pgm db1 pgm

Code	*File*	*Distance*
04	db10 pgm	34638 000000
02	db8 pgm	37396 000000
03	db9 pgm	41736 000000
01	db1 pgm	45870 000000

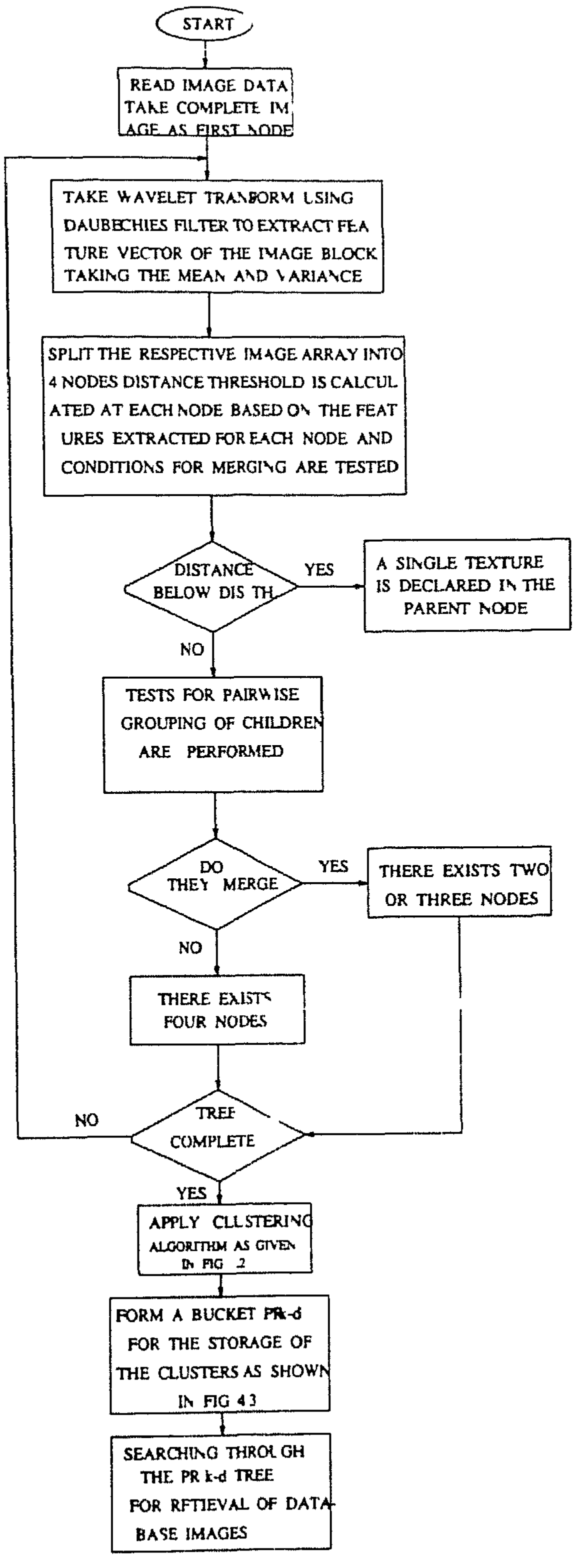


Figure 5 2 Flowchart of the content based indexing and retrieval system

Chapter 6

Conclusion

In this thesis we have presented a method for performing Query by texture on an image database The proposed database is tightly coupled with an image compression algorithm using which feature vectors are constructed and used for clustering and quick searching of the database Texture has been shown to be a visible characteristic of visual information by which image data may be indexed in a large image database The method employed uses a quad tree approach to image segmentation and applies wavelet transformation to extract feature sets from image blocks and derives conditions for merging using distance threshold Without resolving border details between textured regions, we use the homogeneous rectangular blocks of texture within each image to perform indexing in the database This approach in general places no limitations on how features are extracted from image blocks We have used specific feature sets based on the QMF wavelet decomposition because of the discrimination performance and the benefits that this representation offers in a database application The quad tree method offers an efficient approach towards segmenting and representing textures present in the image and provides a general framework by which other discriminating features can be used for image segmentation Clustering method used here is a standard method for grouping data For better clustering linear transformation is applied on the features vectors Our experimental results, given in chapter 5, show that we have obtained desired results of retrieval Referring to the Table 1 and 2 we conclude that the search results point to the segments of the database images that have similar texture information as the query image

6 1 Further Work

As content based indexing has many applications for efficient retrieval of visual information from multimedia databases further research can focus on development of content based indexing techniques which are domain independent and can be automated

Bibliography

[1] Hanan Samet The Quad tree and Related Hierarchical Data Structures' *Computing Surveys* Vol 16 No 2 pp 187 1984

[2] Hanan Samet 'Region Representation Quad trees from Binary Array *Computer Graphics and Image Processing* Vol 13 pp 88 93, 1980

[3] J Guo and A Zhang E Remias, and G Sheikholeslami 'Image Decomposition and Representation in Large Image Database Systems *Journal of Visual Communication and Image Representation* Vol 8 No 2 June, pp 167 181 1997

[4] Aidong Zhang Biao Chang, and Raj Acharya A Fractal Based Clustering Approach in large Visual database Systems *Multimedia Tools and Applications* Vol 3, pp 225 244, 1996

[5] G K Wallace 'The JPEG still picture compression standard , *Communications of the ACM* Vol 34, No 4 pp 30 44 1991

[6] M F Barnsley and A D Sloan A better way to compress images , *BYTE* pp 215 223 1988

[7] Y Fisher and A F Lawrance, 'Fractal image compression for mass storage applications *SPIE Image Storage and retrieval systems* 1662, 1992

[8] Edward Remias G Sheikholeslami, A Zhang Tanveer Fathima Syeda Mahamood 'Supporting Content Based Retrieval in large Image Database Systems *Multimedia Tools and Applications*, Vol 4 pp 153 170 1997

[9] John R Smith and Shih Fu Chang 'Quad tree Segmentation for Texture-Based Image Query' , *Proceedings of ACM Multimedia 94 San Francisco*, Oct 1994 pp 279-286

[10] John R Smith and Shih Fu Chang Transform Features for Texture Classification and Discrimination in Large Image Databases *IEEE international conference on Image Processing Austin Texas* 1984

[11] S Mallat 'A Theory for Multiresolution Signal Decomposition The Wavelet Representation *IEEE Trans Pattern Analysis Machine Intelligence*, Vol 11 pp 674 693 1989

[12] M L Hilton, B D Jawerth and A Sengupta Compressing still and moving images with wavelets *Multimedia systems* Vol 2 No 5 pp 218 227 1994

[13] Keinosuke Fukunaga and Warren L G Koontz A Criterion and Algorithm for Grouping Data", *IEEE Transactions on Computers* pp 108 114 Oct 1970

[14] Keinosuke Fukunaga and Warren L G Koontz Application of the Karhunen Loeve Expansion to Feature Selection and Ordering *IEEE transactions on Computers* Vol C-19, No 4, pp 311 318 April 1970

[15] Keinosuke Fukunaga, 'Introduction to Statistical Pattern Recognition , *Harcourt Brace Javanovich* 1990

[16] S K Chang, C W Yan, Donald C Dimitoff and Timothy Arndt An Intelligent Image Database System *IEEE Trans on Software Engineering* Vol 14, pp 681 688, 1988

[17] J Vass J Yao and X Zhuang University of Missouri Columbia USA

'An Interactive Image Database System *VLDB 98* oct 8-9 1998, pp 53

[18] Giridharan Iyengar and Andrew B Lippman,

"Clustering image using relative entropy for efficient retrieval , *VLDB 98*, oct 8 9,1998, pp 49

[19] S S Wilks

Multidimensional statistical scatter," *Contributions to Probability and Statistics*, I Olkin, Ed Standford, Calif Standford Publishing 1960

[20] H P Friedman and J Rubin,

'On some invariant criteria for grouping data," *Amer Stat Assoc J*, vol 62, pp 1157 1178, December 1967

[21] Hanan Samet,

"The Design and Analysis of Spatial Data Structures" *Addison Wesley publishing company, inc*

Date Slip

This book is to be returned on the date last stamped